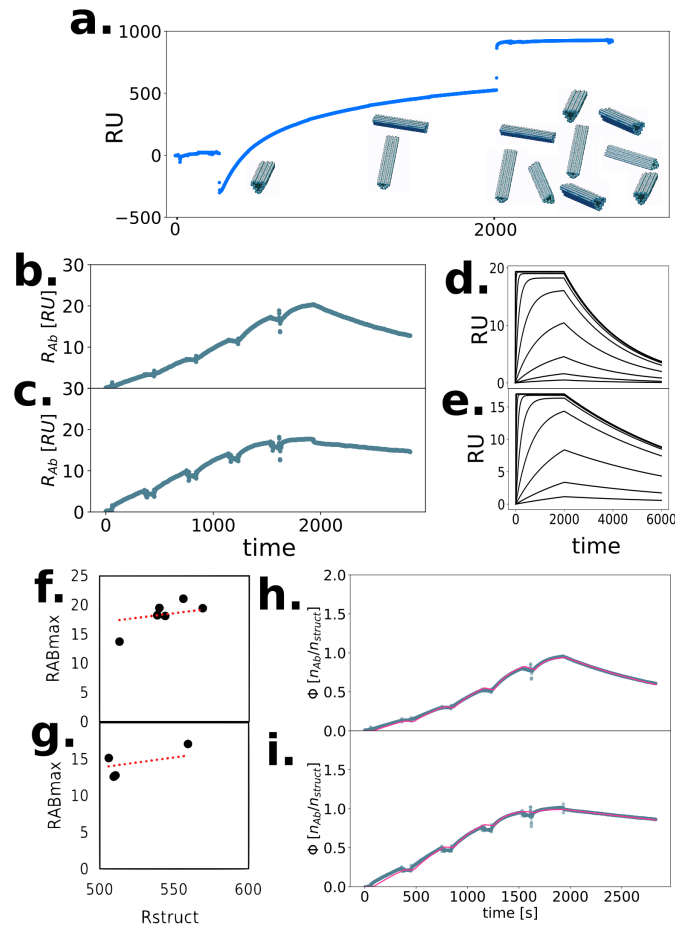

Supplementary information

Stochastic modeling of antibody binding predicts programmable migration on antigen patterns

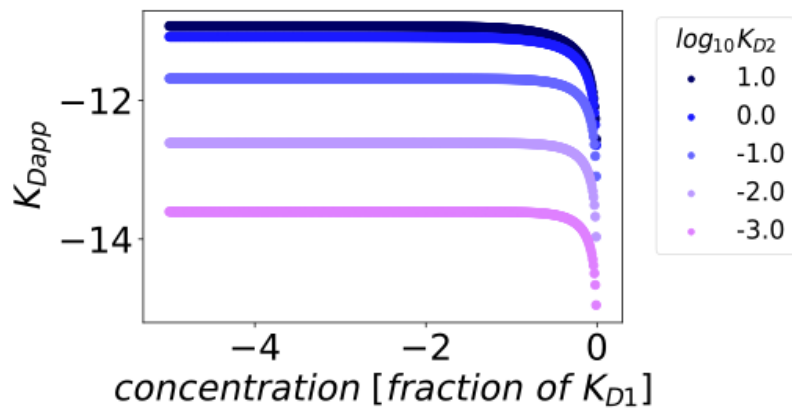
In the format provided by the
authors and unedited

Supplementary Information - Stochastic modeling of antibody binding predicts programmable migration on antigen patterns

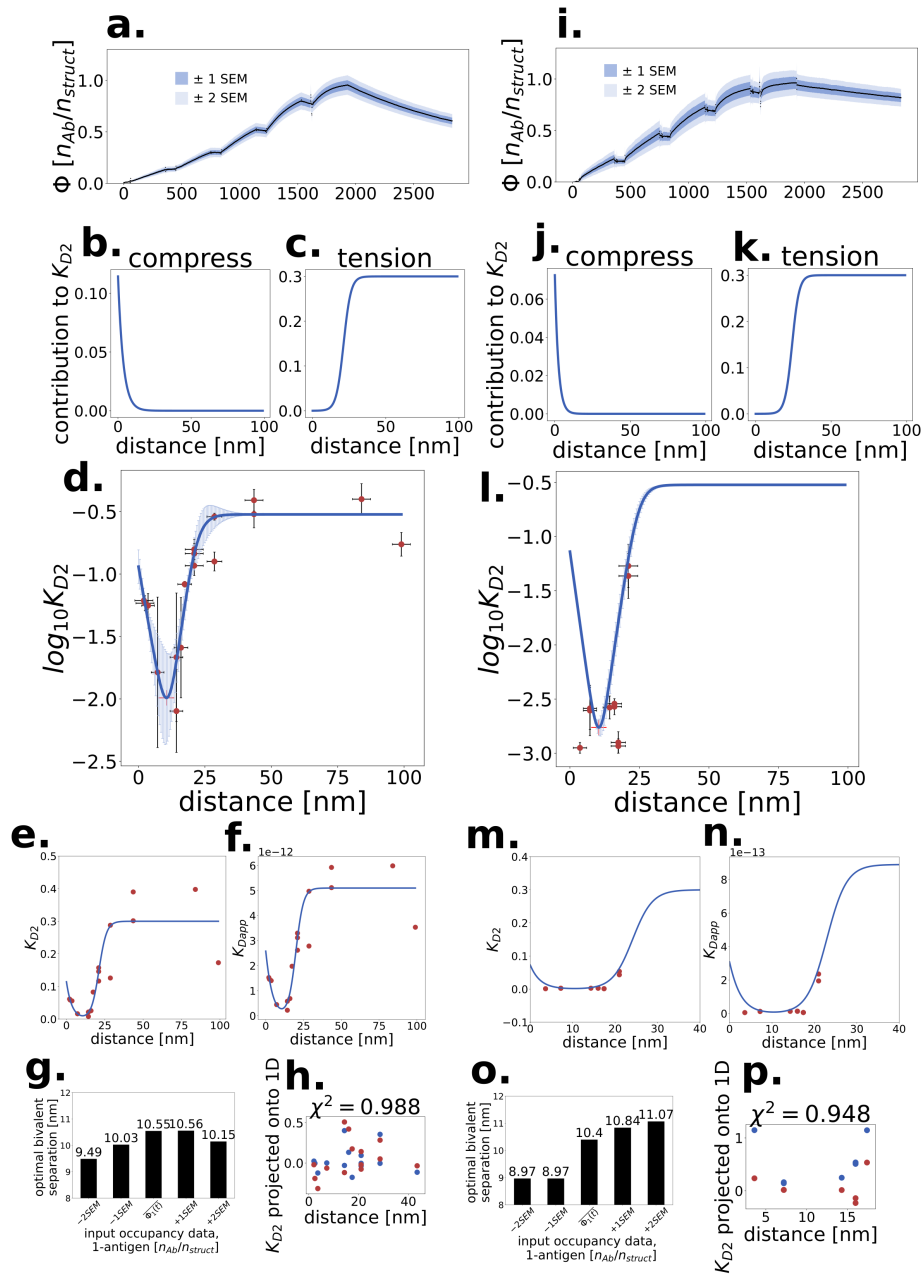
1 Supplementary Figures



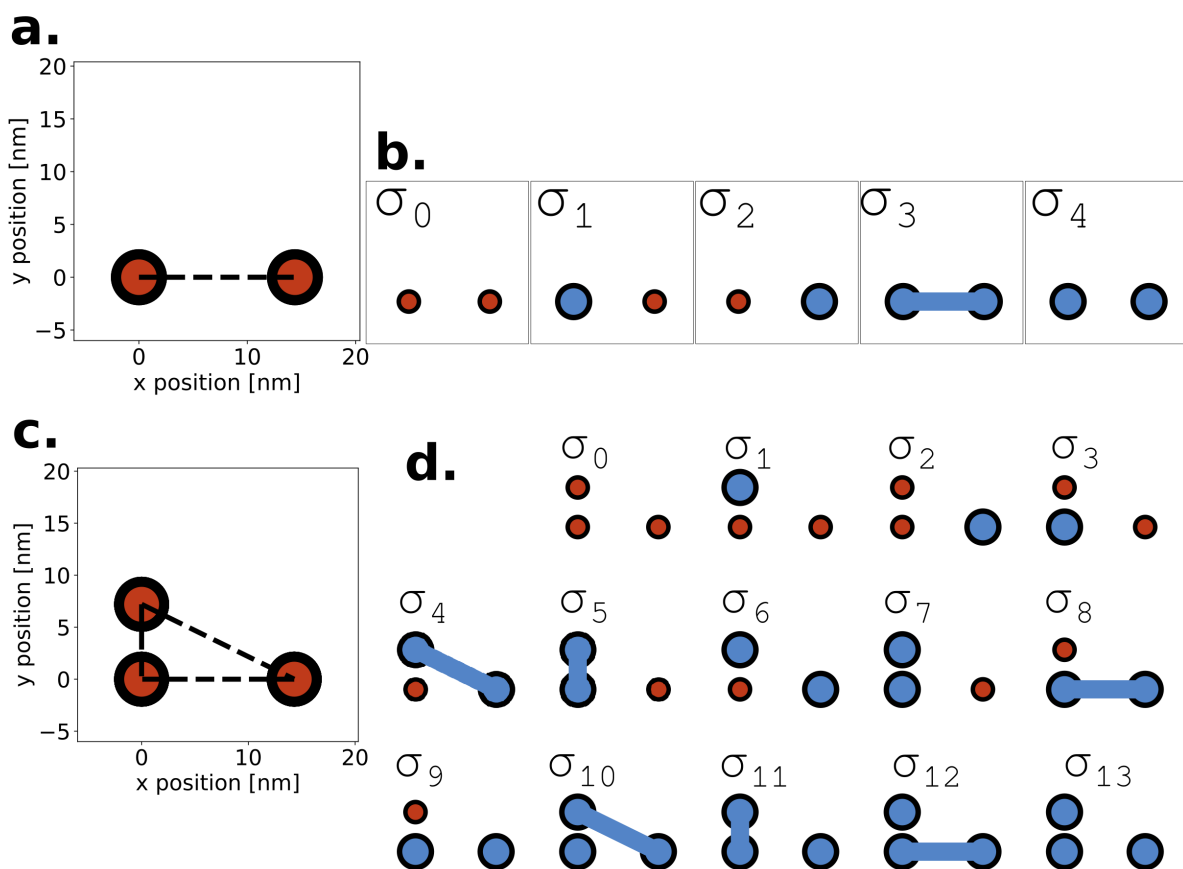
Supplementary Figure 1. a: Sensorgram of DNA origami nanostructure binding to chip in SPR response units (RU). b and c: Sensorgram of antibodies binding to a 1 antigen structure following the structure-binding phase in (a) shown in SPR response units (RU) for rabbit anti-DIG IgG and mouse anti-DIG IgG1 respectively. d and e: Predictions of R_{Ab}^{max} , a point of saturation that the monovalent system converges to as higher concentrations push more structures into the monovalently-bound state for rabbit anti-DIG IgG and mouse anti-DIG IgG1 respectively. f and g: Calculated monovalent R_{Ab}^{max} representing the number of RU corresponding to 1 antibody bound to every structure for various instances of monovalent fits to experimental binding data plotted against their corresponding structure binding signal R_{struct} and the linear fit used to estimate the proportionality of the two variables for rabbit anti-DIG IgG and mouse anti-DIG IgG1 respectively.. h and i: Experimental data (blue) and model (magenta) after normalization by R_{Ab}^{max} , with the more useful units of antibodies per structure (AB/struct) instead of RU for rabbit anti-DIG IgG and mouse anti-DIG IgG1 respectively..



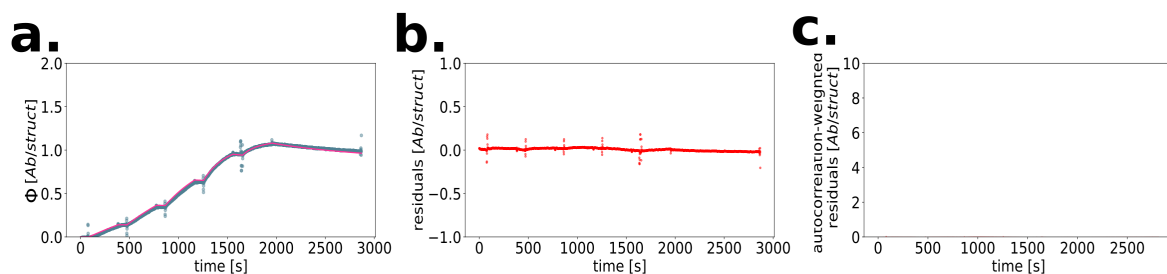
Supplementary Figure 2. Log-log (base 10) plot of the apparent dissociation constant as a function of concentration represented as a fraction of the monovalent dissociation constant. For concentrations approaching that of the monovalent dissociation constant, the apparent binding constant changes significantly, whereas low concentrations lead to less concentration dependence.



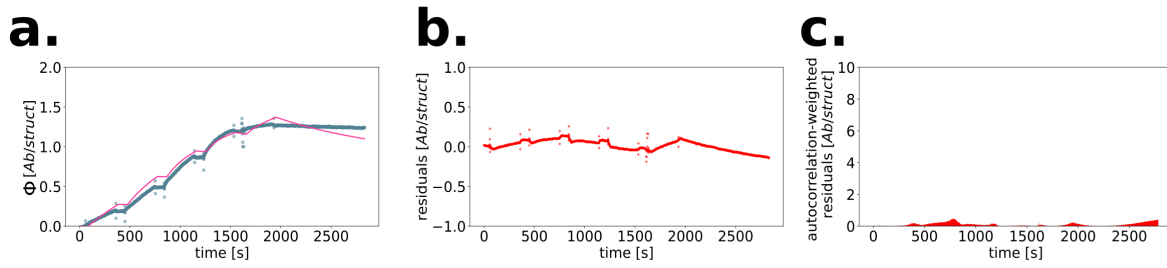
Supplementary Figure 3. Measurement and modeling of spatial tolerance of antibody with increasing spatial separation of 2 antigens rabbit anti-DIG IgG and mouse anti-DIG IgG1 respectively. **a and i:** Occupancy of multiple 1-antigen runs with ribbons indicating the trajectory of a mean run plus or minus 1 or 2 standard error of the mean away for rabbit ($n = 6$) and mouse ($n = 6$) respectively. **b and j:** Exponential compression term of spatial tolerance plotted alone versus distance. **c and k:** Sigmoidal tension term of spatial tolerance plotted alone versus distance. **d and l:** The spatial tolerance function, experimental interconversion constant values (red) and the fitted spatial tolerance model (blue) both plotted against separation distance of two antigens. **e and m:** Spatial tolerance function plotted on non-logarithmic scale. **f and n:** The apparent dissociation constant assuming 1-1 binding kinetics for what is actually a 2-antigen bivalent system calculated from interconversion constant data on the basis of equilibrium occupancy predicted by the bivalent model in terms of individual rate parameters. **g and o:** Sensitivity of the $\min(K_{D2})$ according to the mean 1-antigen input run plus or minus 1 and 2 SEM **h and p:** Goodness of fit of spatial tolerance function visualized as random dispersion of points around a moving average (red) and the model values (blue).



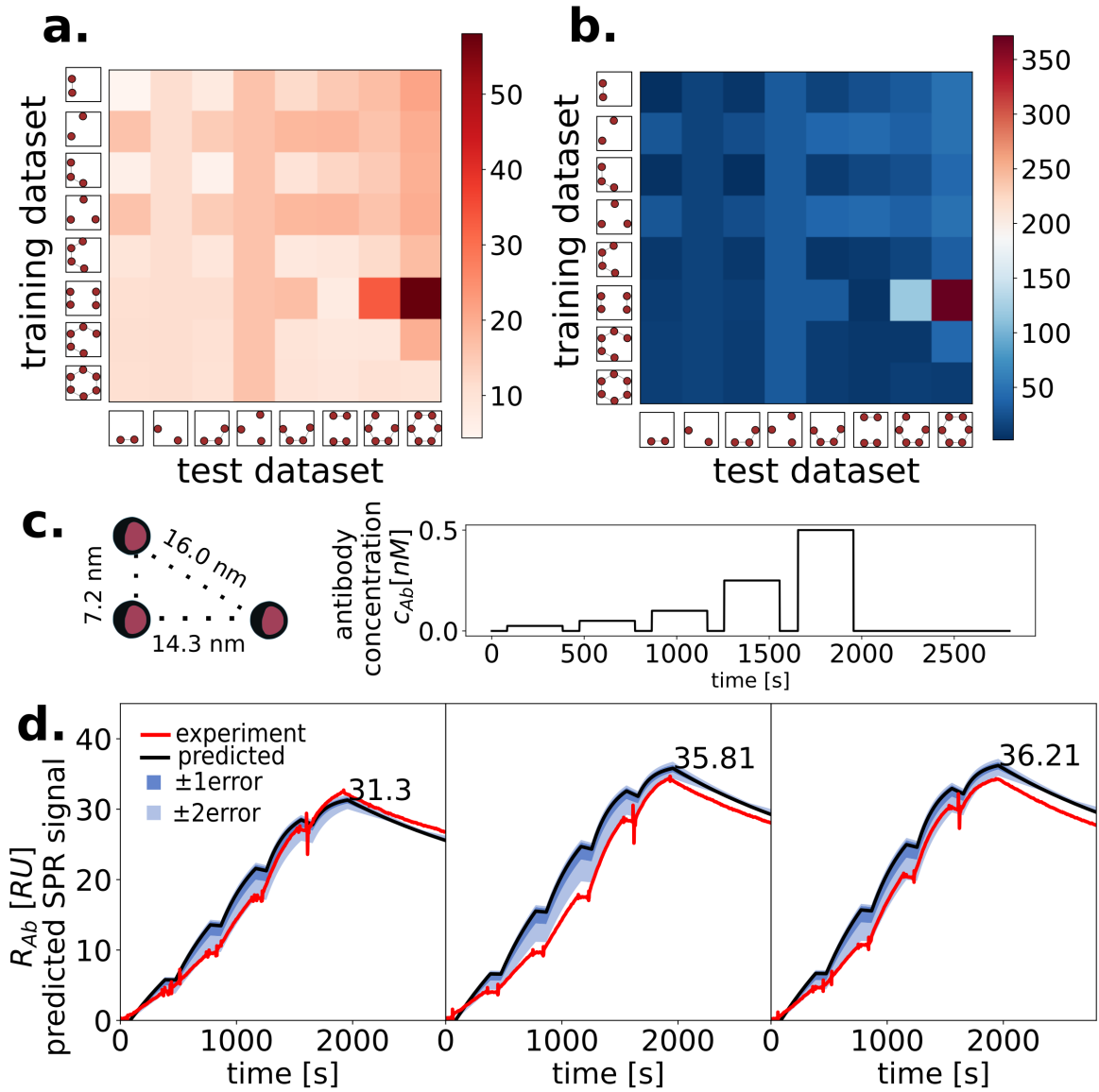
Supplementary Figure 4. Enumerated states for basic and arbitrary geometries. a: Dimensions of a basic bivalent antigen pattern with 15 nm separation between antigens. b: The set of possible states for the basic bivalent geometry with blue dots indicating monovalently bound antibodies, blue lines connecting two dots indicating a bivalently bound antibody, and red dots indicating empty antigen sites. c: The dimensions of an arbitrary geometry, a $12 \times 16 \times 20$ nm right triangle. d: The set of states corresponding to the right triangle geometry.



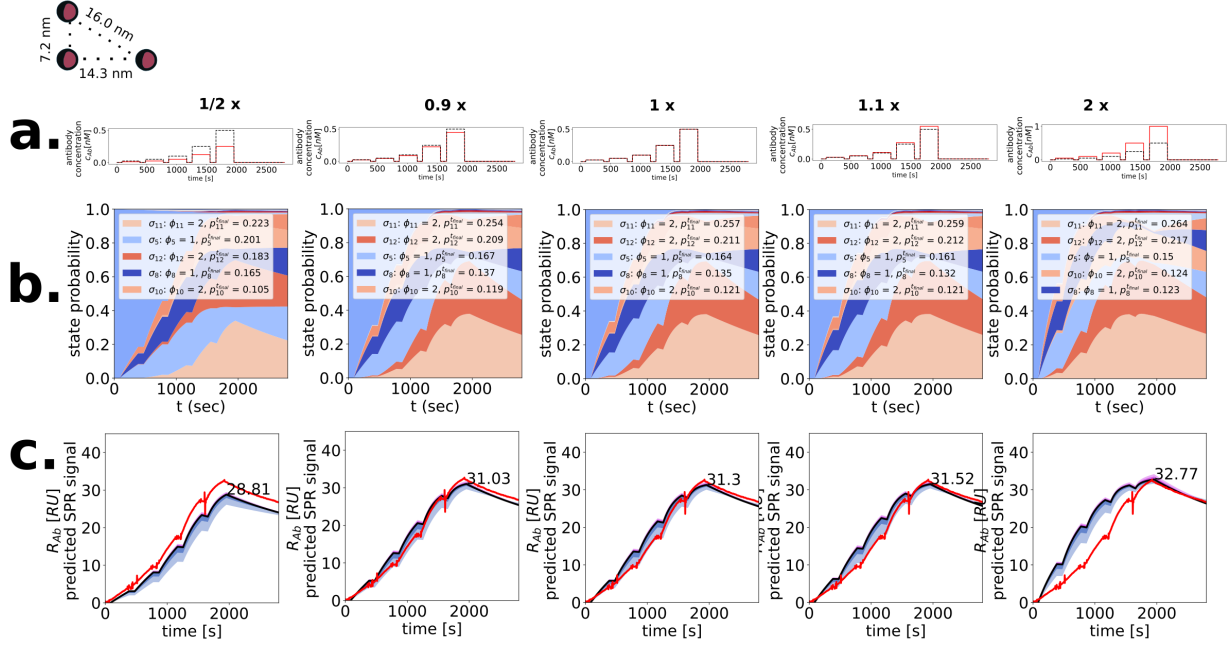
Supplementary Figure 5. Example of a well-parameterized bivalent model, with strong agreement between experimental and theoretical occupancy. a: Markov model (magenta) prediction of average occupancy (antibodies per structure) superimposed on experimental PSPR binding data for a 2 antigen pattern (blue). b: Residuals comparing the theoretical and experimental curves in a. c: Residuals from b weighted by autocorrelation.



Supplementary Figure 6. Example of a misparameterized bivalent model with visual misalignment between theoretical and experimental occupancy curves and corresponding divergence of residuals. a: Markov model (magenta) prediction of average occupancy (antibodies per structure) superimposed on experimental PSPR binding data for a 2 antigen pattern (blue). b: Residuals comparing the theoretical and experimental curves in a. c: Residuals from b weighted by autocorrelation.



Supplementary Figure 7. a: Matrix depicting the sum of residuals (color map) for Markov models parameterized on experimental binding data from antigen patterns on the vertical axis (training dataset) and performance-tested against other patterns (test dataset). b: Same as (a) except that colormap indicates the residuals weighted by their autocorrelations. c: Antigen configuration (left) and run design (right) for a blinded evaluation of model predictive power. d: Triplicate set of independent PSPR trials with predictions based on parameterized model (black) compared with experimental outcomes (red).



Supplementary Figure 8. Sensitivity of predicted SPR curve to concentration changes. a: Injection/dosing schemes for the single cycle kinetics program multiplied by various factors (red) compared with the original scheme (black). b: The resulting transient probability distribution predicted from modified concentration schemes. c: The resulting comparison of predicted average occupancy to that of the experimental validation under different concentration multipliers. From left to right, concentration multipliers are 0.5, 0.9, 1, 1.1, and 2 x.

2 Supplementary Algorithms

Algorithm 1 Breadth first search enumeration of states

```

1: function ENUMERATE( $\mathbf{D}, k_1, k_{-1}, k_2, k_{-2}$ )  $\triangleright$  Distance matrix of antigen pattern supplied as input
2:    $bfsq \leftarrow$  empty list
3:    $visited \leftarrow$  empty list
4:   append  $\sigma_0$  to  $bfsq$   $\triangleright$  Initialize a search queue with the first (empty) state i.e. no antibodies
5:   while length( $bfsq$ )  $\neq 0$  do
6:     get current state  $\sigma_i$  from top of  $bfsq$  queue
7:     for site  $\zeta_k$  do  $\triangleright$  Where  $k = 0, 1, 2, \dots$  number the sites in the antigen pattern
8:       get binding status of site  $\zeta_k$  (empty, monovalent, bivalent) from  $\sigma_i$ 
9:       if site  $\zeta_k$  is monovalent then
10:        for neighboring site  $\zeta_k \forall s$  such that  $\mathbf{D}_{k,s} \leq d_{crit}$  do
11:          if  $\zeta_j$  is empty and no obstructions then  $\triangleright$  e.g. intersection with other antibody
12:            Let  $\sigma_j$  be the new adjacent state created by a bivalent transition from  $\sigma_i$ 
13:             $\zeta_k \leftarrow$  bivalent
14:             $\zeta_s \leftarrow$  bivalent
15:            if  $\sigma_j \notin visited$  then
16:              append  $\sigma_j$  to  $bfsq$ 
17:               $\Lambda_{i,j} \leftarrow k_2$ 
18:               $\Lambda_{j,i} \leftarrow k_{-2}$ 
19:            end if

```

```

20:         end if
21:     end for
22:     Let  $\sigma_j$  be the new adjacent state created by a monovalent removal from site  $\zeta_k$ 
23:      $\zeta_k \leftarrow empty$ 
24:     if  $\sigma_j \notin visited$  then
25:         append  $\sigma_j$  to bfsq
26:          $\Lambda_{i,j} \leftarrow k_{-1}$ 
27:          $\Lambda_{j,i} \leftarrow k_1$ 
28:     end if
29:     else if site  $\zeta_k$  is bivalent then
30:         Let  $\sigma_{j1}$  and  $\sigma_{j2}$  be the two possible monovalent states formed from  $\sigma_i$ 
31:          $\sigma_{j1} : \zeta_k \leftarrow empty$  and  $\zeta_s \leftarrow monovalent$   $\triangleright$  Where  $\zeta_s$  is the bivalent partner site of  $\zeta_k$ 
32:          $\sigma_{j2} : \zeta_k \leftarrow monovalent$  and  $\zeta_s \leftarrow empty$   $\triangleright$  Where  $\zeta_s$  is the bivalent partner site of  $\zeta_k$ 
33:         if  $\sigma_{j1} \notin visited$  then
34:             append  $\sigma_{j1}$  to bfsq
35:              $\Lambda_{i,j} \leftarrow k_{-2}$ 
36:              $\Lambda_{j,i} \leftarrow k_2$ 
37:         end if
38:         if  $\sigma_{j2} \notin visited$  then
39:             append  $\sigma_{j2}$  to bfsq
40:              $\Lambda_{i,j} \leftarrow k_{-2}$ 
41:              $\Lambda_{j,i} \leftarrow k_2$ 
42:         end if
43:     else if site  $\zeta_k$  is empty then
44:         Let  $\sigma_j$  be the new adjacent state created by a monovalent addition to site  $\zeta_k$ 
45:          $\zeta_k \leftarrow monovalent$ 
46:         if  $\sigma_j \notin visited$  then
47:             append  $\sigma_j$  to bfsq
48:              $\Lambda_{i,j} \leftarrow k_1$ 
49:              $\Lambda_{j,i} \leftarrow k_{-1}$ 
50:         end if
51:     end if
52: end for
53: remove  $\sigma_i$  from bfsq
54: end while
55: return  $\Lambda$ 
56: end function

```

end

Algorithm 2 *Uniformization of transition rate matrix*

```

1: function UNIFORMIZE( $\Lambda$ )  $\triangleright$  Transition rate matrix is supplied as input
2:    $\mathbf{Q} \leftarrow zeros(N, N)$ 
3:   for  $i = 0$  to  $N$  do
4:     for  $j = 0$  to  $N$  do
5:       if  $i = j$  then
6:          $\mathbf{Q}_{i,j} \leftarrow -1 \cdot sum(\Lambda_i)$ 
7:       else
8:          $\mathbf{Q}_{i,j} \leftarrow \Lambda_{i,j}$ 
9:       end if
10:    end for
11:  end for
12:   $q = max(sum(\Lambda, axis = 1))$ 

```

```

13:    $\mathbf{U} \leftarrow \mathbf{I}(N) + \mathbf{Q}/q$ 
14:   return  $\mathbf{U}$ 
15: end function

```

end

Algorithm 3 *Simulating transient evolution of binding states during a PSPR run*

```

1: function SPRRUN( $\mathbf{p}(t_0), \mathbf{U}$ )           ▷ An initial probability distribution  $\mathbf{p}(t_0)$  is supplied as input
2:    $\mathbf{p} \leftarrow \text{zeros}(N, t_{max})$ 
3:    $\mathbf{p}(t = 0) \leftarrow \mathbf{p}(t_0)$ 
4:    $\Delta t \leftarrow 1 \text{ sec}$ 
5:    $\eta_{max} \leftarrow 15$                  ▷ The iteration depth to approximate the series from Equation 79
6:   for  $t = t_0$  to  $t_{max} - 1$  do
7:      $\mathbf{p}(t + \Delta t) \leftarrow \text{zeros}(N)$ 
8:     for  $\eta = 0$  to  $\eta_{max}$  do
9:        $\Psi \leftarrow (e^{-q\Delta t}(q\Delta t)^\eta)/\eta!$            ▷ Where  $\Psi$  is a partial summation from Equation 79
10:      if  $\eta = 0$  then
11:         $\pi_i \leftarrow \mathbf{p}(t)$                                ▷  $\pi_i$  is an intermediate variable
12:      else
13:         $\pi_i \leftarrow \pi_i \cdot \mathbf{U}$ 
14:      end if
15:       $\mathbf{p}(t + \Delta t) \leftarrow \mathbf{p}(t + \Delta t) + \pi_i \cdot \Psi$ 
16:    end for
17:  end for
18:  return  $\mathbf{p}$ 
19: end function

```

end

Algorithm 4 *Monte Carlo simulation of antibody binding dynamics on patterned antigens*

```

1: function RANDOMWALK( $\sigma_0$ )           ▷ Supply an initial condition, i.e. starting state
2:    $t \leftarrow t_0$ 
3:    $\sigma_t \leftarrow \sigma_0$ 
4:   while  $t \leq t_{max}$  do
5:      $\lambda_{\text{local}}, \sigma_{t_{n+1}} \leftarrow \text{GetLocalExitArray}(\sigma_t)$    ▷ i.e. search current state for valid transitions
6:      $\lambda_{\text{exit}} \leftarrow \sum_{\forall i} \lambda_{\text{local}}$ 
7:      $p_{\text{exit}} \leftarrow \text{random}(0, 1)$            ▷ get a random number between 0 and 1
8:     if  $\lambda_{\text{exit}} = 0$  then           ▷ A sink state, e.g. if concentration is zero and pattern is empty
9:        $\tau \leftarrow t_{max} - t$            ▷ The dwell time of this state extends to the end of the simulation
10:    else
11:       $\tau \leftarrow \frac{\ln(1/p_{\text{exit}})}{\lambda_{\text{exit}}}$            ▷ dwell time for the current state  $\sigma_t$ 
12:       $\sigma_t \leftarrow \text{RandomChoice}(\sigma_{t_{n+1}})$            ▷ Where  $p_{\sigma_{n+1}} = \lambda_{\text{local}}/\lambda_{\text{exit}}$ 
13:    end if
14:     $t \leftarrow t + \tau$ 
15:  end while
16: end function

```

end